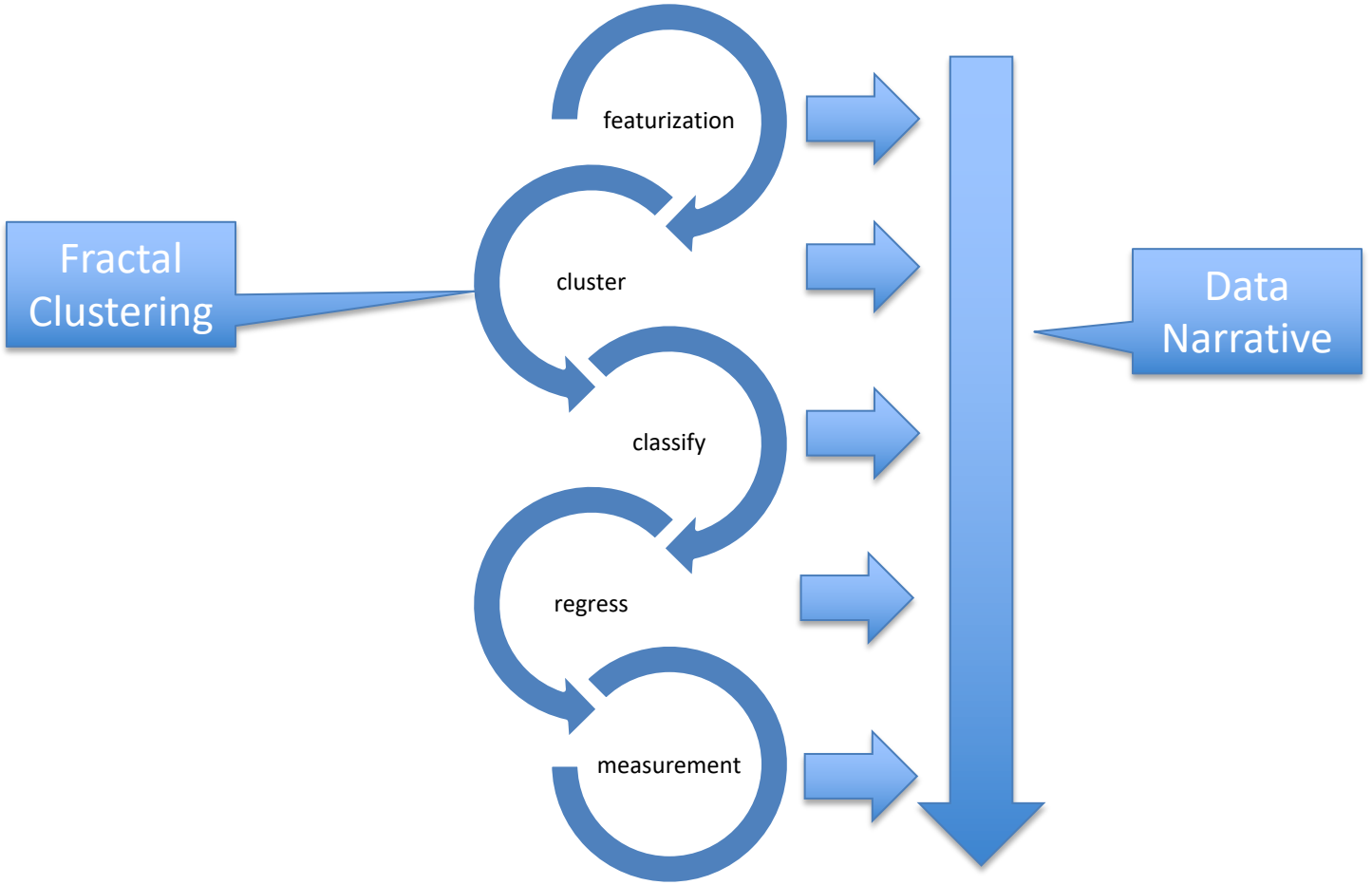
Session 13 – ML-Based Software Engineering

Dr Arsanjani

The ML Life-cycle is a Journey of Increased Refinement

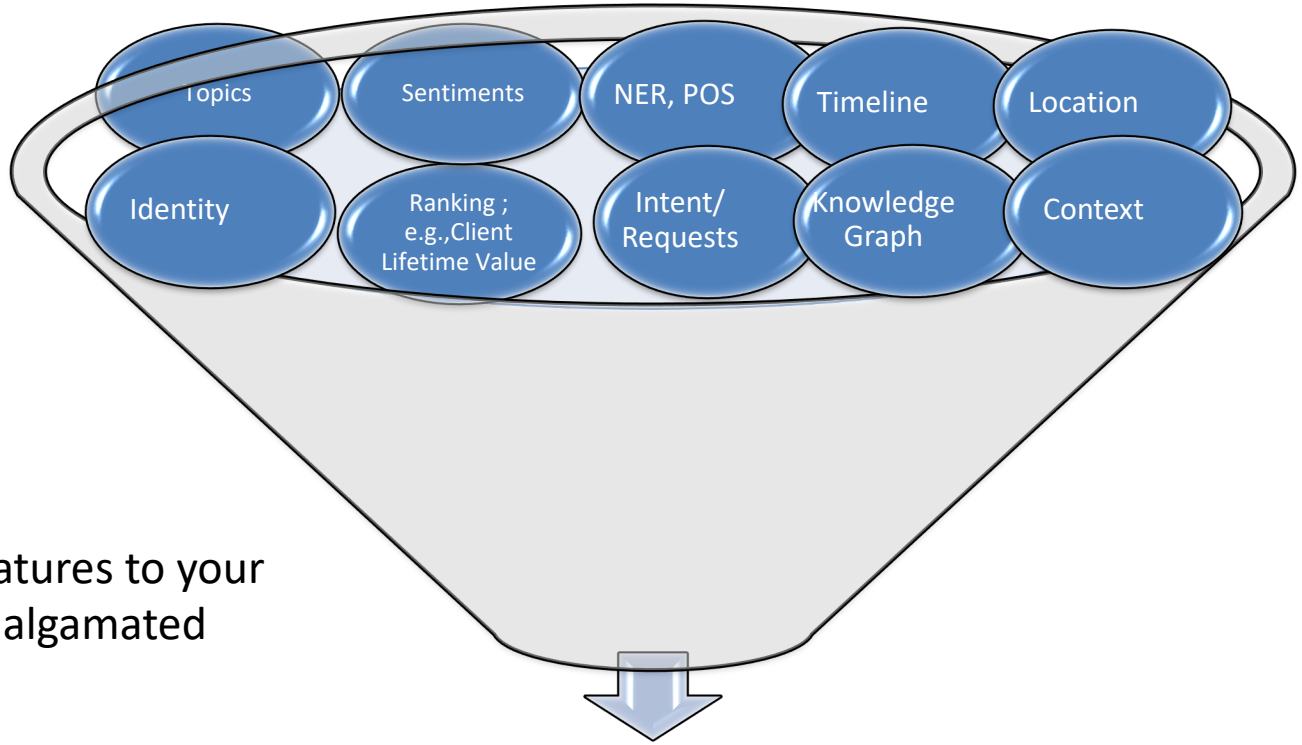


Sprint 4 : Towards the end/beginning of the notebook

	Algo/Accuracy	Micro-factors	Confusion Matrix	Articles
Dataset 1	Muller loop RBF SVM .45			
Amalgamated with dataset2	Muller loop RBF SVM .55			
Converted True/.false to numeric	.30 → .70			

Distillations

NLP



Add new features to your existing, amalgamated dataset

Distilled Contextual Embedding

Add more accuracy, precision, recall, f1 , RMSE, CM

Data Sets

Original DataSet

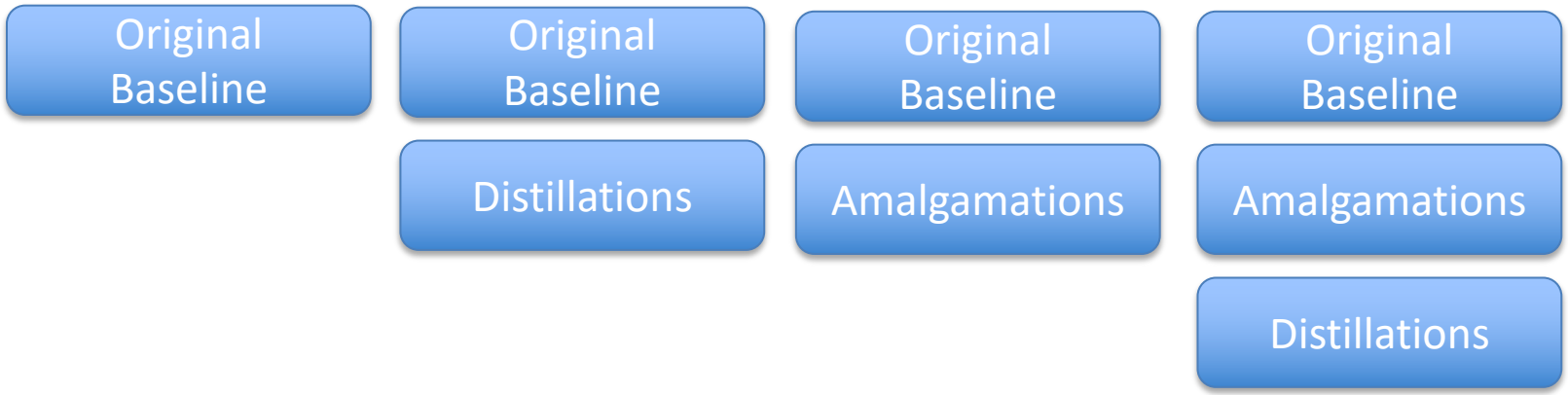
- Find a canonical that fits best as a reference, a baseline

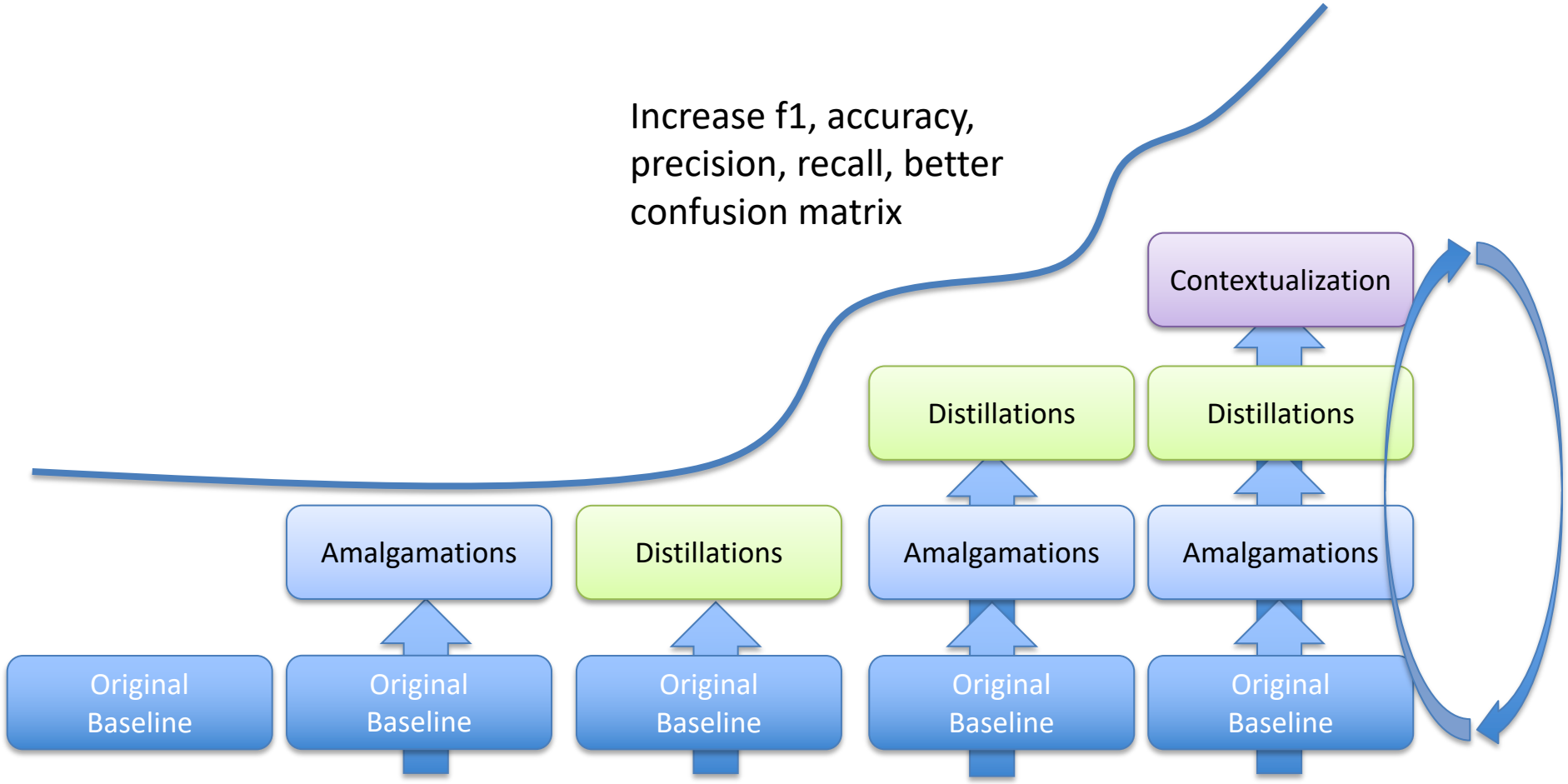
Amalgamations

- Data Set 2 (Scraped?)
- Data Set 3 (Scraped?)

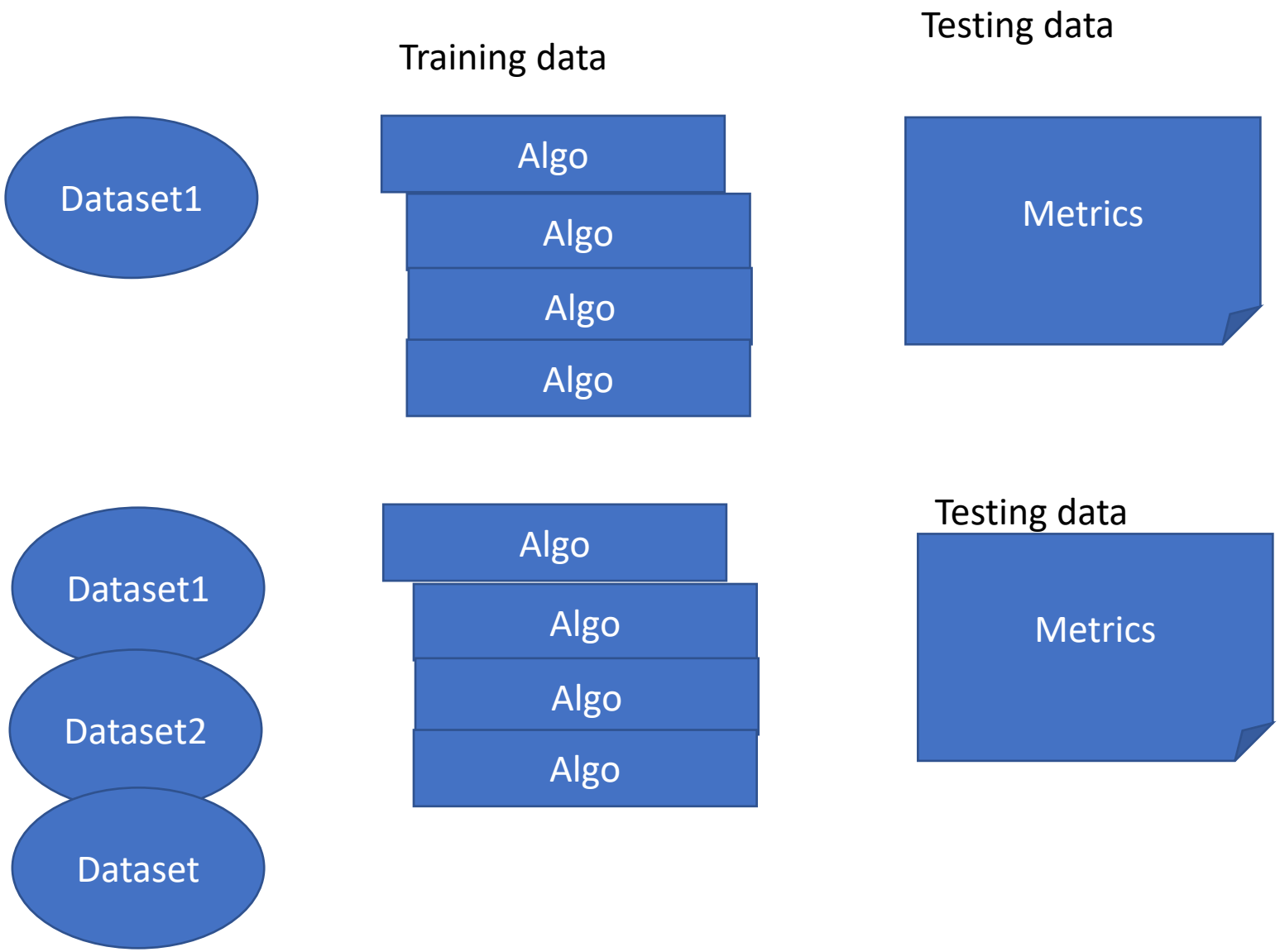
Distillations

Add distillations to original dataset or other datasets, whichever more promising


Original Baseline
Original Baseline
Distillations
Original Baseline
Amalgamations
Original Baseline
Amalgamations
Distillations


Increase f1, accuracy, precision, recall, better confusion matrix
Contextualization
Distillations
Distillations
Amalgamations
Distillations
Amalgamations
Amalgamations
Original Baseline
Original Baseline
Original Baseline
Original Baseline
Original Baseline

- **Top Ten Criteria** to keep in mind (re-iterated from classes): (edited)
- 0. Can Dr.Arsanjani and Sri run my notebook? Is my data all in the shared folders?
  Each person must submit their work separately (even if you just copy the submission document from your team of say four and it has all four of your urls in it)
  This way we can actually go down the list of students and grade. (edited)
- Did I use:
- a. Use Imports for each factor,
- b. one class for each factor (edited)
- 2. [Pipelines]
- 3. Pickl my models and load if I have already trained them. Pls note: When you run code that takes a lot of time, pickl the results : for example: why? so next time anyone runs it, you can check if the file exists, if yes, then pick.load() instead of retraining or instead of going through that loop for 30 minutes !!! (
- 4. used at least 3 amalgamations : e.g., liar liar + twitter + fake news. Make sure you provide links to the datasets you use for training, testing and validation , before and after amalgamations.
  a. test, train validation --> url (googledrive), source --> url
  b. amalgamation dataset 2 --> url, result of amalgamation --> url (googledrive)
  c. amalgmation dataset 3 --> url, result of amalgamation --> url (ggoogle drive) (edited)
- 5. used multiple algorithms. Compare results in a table : with precision, recall, accuracy and f1 listed. (edited)
- 6. shown precision, recall, accuracy and f1,
- 7. confusion matrix (display actual matrix vs just printing it) (edited)
- 8. interpret the results and show them in a table: how did I get better results from amalgamation and distillation? Is the shape of data influencing the results? How? (edited)
- 9. Used our machine learning lifecycle (posted on the website) up to step 9, deep learning
- 10. In my video and submission text have I shown what I have done beyond last week, or weeks before (don't just go over the same things you did in the last video!)


Training data
Testing data
Dataset1
Algo
Algo
Algo
Algo
Metrics
Dataset1
Dataset2
Dataset
Algo
Algo
Algo
Algo
Testing data
Metrics

Make your ML code Modular

- Google:
  - How to implement classes in python
  - How to use “import” in colab
  - . ipynb → .py load it in the folder next to your notebook

- Team name
- Student Name
- Colab Url
- Data 1 Url, Data 2 Url Data 3 Url
- Description:
  - Improve since last iteration
- Video : describe the above , and only what you have improved
- Or the delta since the last sprint

Software Engineering for ML

Research : comparisons with tables and citations of githubs or articles that influenced you

import

Data and folders

Comparing in a table as you make progress

Pickle your models

Load and infer

AV Project: Colab for Code

- Data in common folder
- Class for each Factor
- Class Factor()
  - Prep
    - Def Read the csv(s) : (def a func!!)
      - Access the common googledrive
      - LL
      - FNC
      - Your data set for your factor
    - Data Preparation depends on the algorithm
      - Some algo with numbers
      - Some work with text/unstru (MNBayes)
      - BERT : string → delimiters in the string
      - Categ → one hot encode
      - Numeric → leave them!
      - LDA
    - Algos: Muller loop or additional Transformers
    - Metrics ? To measure model perf
      - F1, recall, precision, accuracy, confusion matrix (print out)

Alg Comparisons based on Metrics

	MNB	XGB	LR	DNN
accuracy				
prec				
Recall				
f1				

You can run individually or use a PIPELINE!!!

Combine the factors and get a score

- Import each factor
- Instantiate them
- Call predict on each
- Method 1
  - Polynomial (Nov 24)
  - Model-based (Dec 1)
    - Group Model –based
    - Combine all teams polynomial (Dec 1)
      - Team Coordinator nominated
      - Team coordinators get together before Dec 1 , and spend an hour combining the code into one NB

Polynomial aka method 1

```
def getAVTruthScore(text = source)
# Using normalized accuracy as weights
weight = [0.84, 0.56, 0.95,0.7,0.13] # using the (normalized) accuracy as weigths
#[0.5 CREDIBILITY AND RELIABILITY, 0.3, 0.1,0.1,0.13]
#normalize the weight matrix between 0..1
w = [float(i)/sum(weight) for i in weight]
sumW = 0
prob = []

if (news!=""):
'''
CREDIBILITY AND RELIABILITY
'''
prob.append(w[0]*credreliable.DATAMINERS_getCredRelScore(source))
sumW+=w[0]

probTotal = sum(prob[0:len(prob)]) / sumW
return probTotal
```

- Train
  - Pickle the model (dump), consider tar.gz.
    - You may want to tar gz if cloud deployment
- Inference, Serving, Prediction
  - Pickle: read (load)
  - myFeature = pickle.load(“factor_01_word_freq.pkl”)
  - myFeature.predict(article, [], [])

Data : GoogleDrive

- AlternusVeraDataSets2020
  - <Team name folder> Avengers
    - LL
    - <Factor –> Misleading Intentions
      - “Dataset”
      - “Model”
        - Avengers_Factor_TitleVsBody.pkl
        - Avengers_Factor_ LDA Topic Features
        - Avengers_Factor_ Text Rank
        - Avengers_Factor_ Misleading Intentions

Team Avengers

- Title vs body
- LDA Topic Features
- Text Rank
- Misleading Intentions

Pickle → file what filename?

TeamName_”Factor”_<factorname>.pkl
Avengers_Factor_TitleVsBody.pkl

Where does my source go?
Colab but be modular!!

- Colab
- From factors Import team_avengers_factor_title_vs_body as tvb
- team_avengers_factor_title_vs_body.py
- Similar to Import pandas as pd
- E.g., pypl: https://pypi.org/project/gensim/

- Import teamname_factor as team_factor